Hepatoprotective effects of systemic ER activation

BulkRNAseq - Transcriptome molecular signatures

Christian Sommerauer & Carlos Gallardo

11 August, 2022

```
# source and library import
source('code/00_helper_functions.R')
library(tidyverse)
library(DESeq2)
```

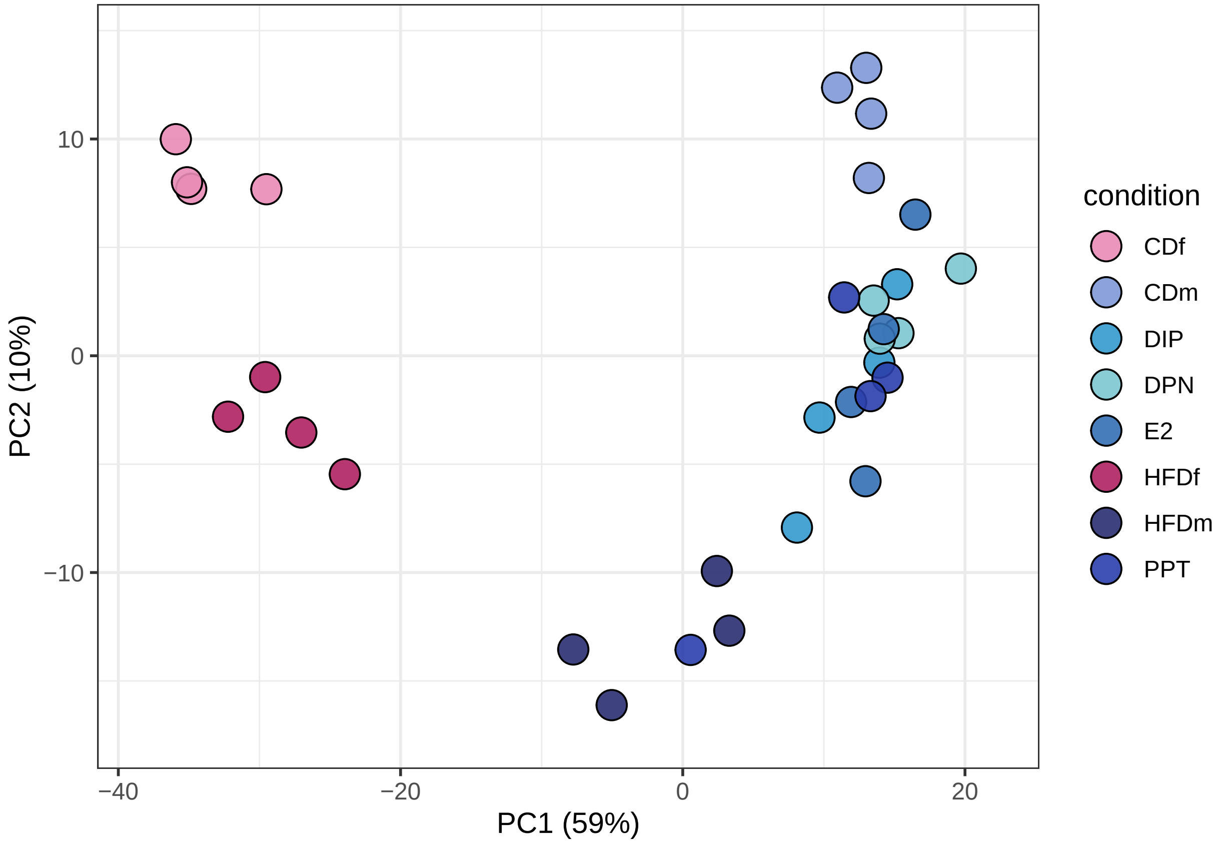
```
# color palettes
colPals <- list()
colPals$conditions <- setNames(c('#E98BB6', '#B02262', '#7F9AD7', '#2A2F72', '#7DC7D1', '#339ACD', '#35
                               c('CDf', 'HFDf', 'CDm', 'HFDm', 'DPN', 'DIP', 'E2', 'PPT'))
colPals$RdBu <- rev(RColorBrewer::brewer.pal(n=11, name = 'RdBu'))
colPals$UpDown <- setNames(colPals$RdBu[c(10,2)],
                           c('up', 'down'))
```

Load data

```
# consensus differentially expressed genes
DEGs <- readRDS('results/bulkRNAseq_mmus_DEGs.rds')

# raw counts RNAseq
raw_counts <- read.table(
  file = 'data/bulkRNAseq_mmus_rawcounts.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE) %>%
  dplyr::filter(geneID %in% DEGs$unfilt$CDfVsCDm$ensembl_gene_id) %>%
  tibble::column_to_rownames('geneID') %>%
  as.matrix()

# gene lengths
gene_len <- read.table(
  file = 'data/bulkRNAseq_mmus_gene_lengths.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE) %>%
  dplyr::filter(geneID %in% DEGs$unfilt$CDfVsCDm$ensembl_gene_id)

# design RNAseq
design_meta <- read.table(
  file = 'data/bulkRNAseq_mmus_design.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE)

# ensembl gene annotation (Mus musculus)
gene_ann <- read.table(
  file = 'data/ensembl_mmus_sep2019_annotation.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE,
  fill = FALSE,
  quote = '') %>%
  dplyr::filter(ensembl_gene_id %in% DEGs$unfilt$CDfVsCDm$ensembl_gene_id) %>%
  dplyr::arrange(factor(ensembl_gene_id, levels = rownames(raw_counts)))
```

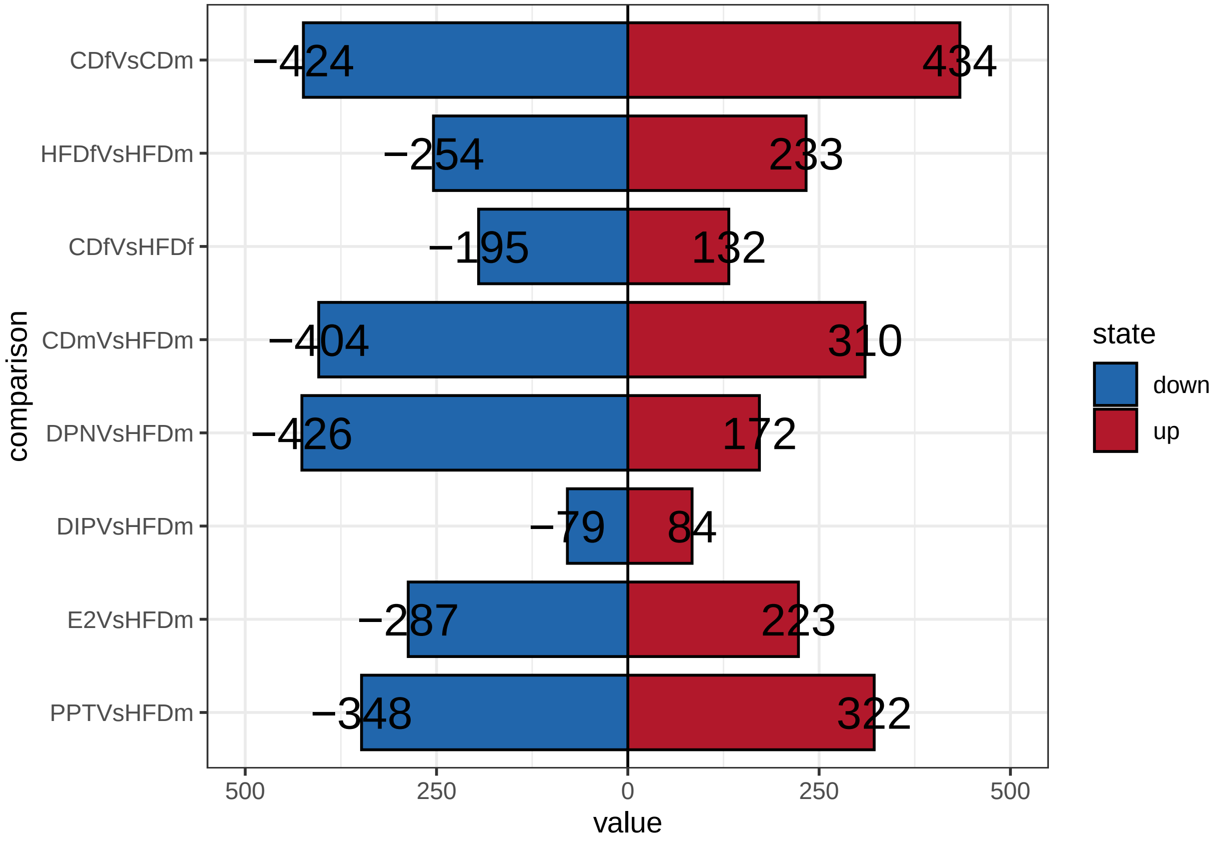
Principal component analysis (PCA)

```
pca_res <- DESeq2::DESeqDataSetFromMatrix(countData = raw_counts,
                                          colData = design_meta,
                                          design = ~0 + condition) %>%
  DESeq2::estimateSizeFactors() %>%
  DESeq2::DESeq() %>%
  DESeq2::vst(blind = FALSE) %>%
  assay() %>%
  doPCA()

df <- data.frame(PC1 = pca_res$pcs$PC1,
                 PC2 = pca_res$pcs$PC2,
                 condition = design_meta$condition)

ggplot(df, aes(x=PC1, y=PC2, fill=condition),) +
  geom_point(shape=21, size=5, stroke=0.5, color='black') +
  scale_fill_manual(values = alpha(colPals$conditions, 0.9)) +
  scale_x_continuous(expand = expansion(mult = c(.1, .1))) +
  scale_y_continuous(
    expand =expansion(mult = c(.1, .1))) +
  xlab(paste0('PC1 (', round(pca_res$perc_var[1]), '%)')) +
  ylab(paste0('PC2 (', round(pca_res$perc_var[2]), '%)')) +
  theme_bw()
```



Differentially expressed genes (DEGs)

```
up <- lapply(DEGs$filt, function(x) sum(x$log2FoldChange>0)) %>% unlist()
down <- lapply(DEGs$filt, function(x) sum(x$log2FoldChange<0)) %>% unlist()

df <- data.frame(comparison=factor(rep(names(DEGs$filt),2), levels=names(DEGs$filt)),
                 state=c(rep('up', length(up)), rep('down', length(down))),
                 value=c(up, down*-1))

ggplot(df, aes(x=comparison, y=value, fill=state, label=value)) +
  geom_hline(yintercept = 0, linetype='solid', size=0.5) +
  geom_bar(color='black', size=0.5, width=0.8, position='stack', stat='identity') +
  geom_text(size=6) +
  scale_fill_manual(values = colPals$UpDown) +
  scale_x_discrete(limits = rev) +
  scale_y_continuous(limits = c(-500,500), labels = c(500,250,0,250,500)) +
  coord_flip() +
  theme_bw()
```

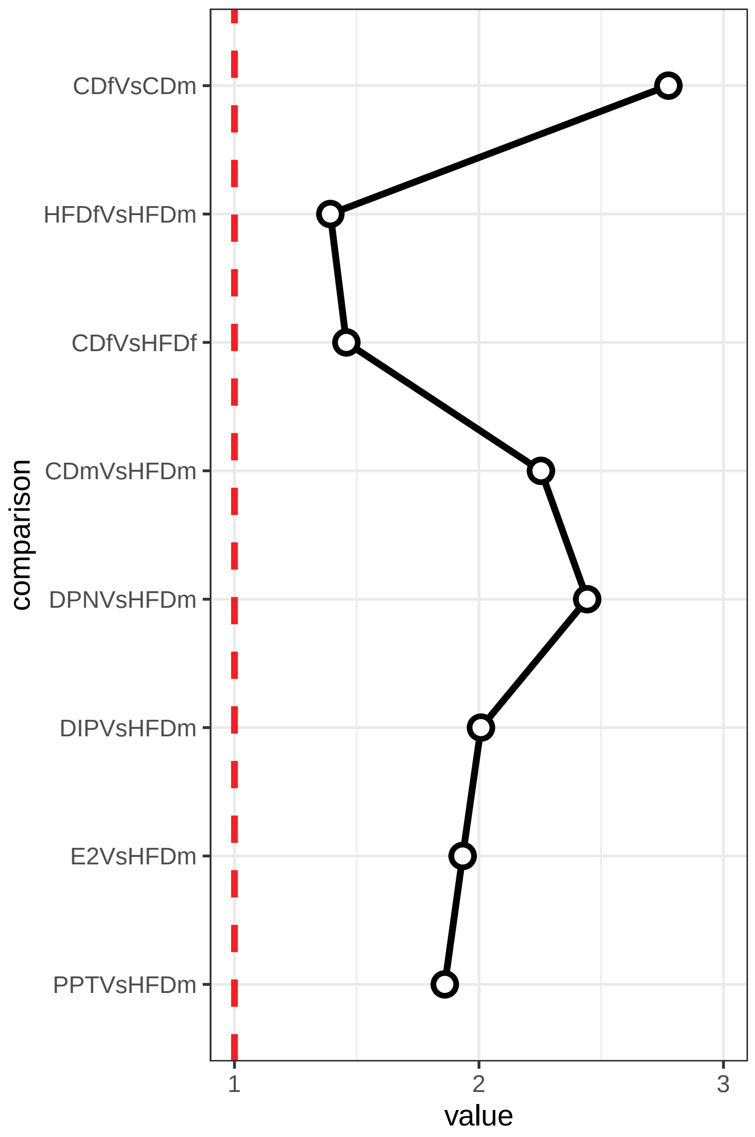


Filter and normalize RNAseq data

```
RNAseq <- list()

# remove outlier sample PPT_HFD_male_4
RNAseq$counts <- raw_counts %>%
  as.data.frame() %>%
  dplyr::select(-PPT_HFD_male_4)

RNAseq$design_meta <- design_meta %>%
  dplyr::filter(sample != 'PPT_HFD_male_4')

RNAseq$annotation <- gene_ann %>%
  dplyr::rename(geneID = ensembl_gene_id) %>%
  dplyr::left_join(gene_len, by = 'geneID')

RNAseq$cpm <- RNAseq$counts %>%
  normalizeData(method = 'CPM')

RNAseq$tpm <- RNAseq$counts %>%
  normalizeData(len = RNAseq$annotation$length, method = 'TPM')
```

Transcriptome-wide signal-to-noise ratios (tSNR)

```r
df <- RNAseq$tpm %>%
  scaleData(method = 'zscore') %>%
  tSNR(group.lbls = RNAseq$design_meta$condition) %>%
  tibble::rownames_to_column(var = 'X') %>%
  tidyr::pivot_longer(cols = dplyr::everything()[-1], names_to = 'Y') %>%
  tidyr::unite(col = 'comparison', X, Y, sep = 'Vs') %>%
  dplyr::filter(comparison %in% names(DEGs$filt)) %>%
  dplyr::mutate(comparison=factor(comparison, levels = names(DEGs$filt)))

ggplot(df, aes(x=comparison, y=value)) +
  geom_line(group=1, size=1.2) +
  geom_point(shape=21, size=3, stroke=1.5, color='black', fill='white') +
  geom_hline(yintercept = 1, linetype='dashed', size=1.2, color='#EF2126') +
  scale_x_discrete(limits = rev) +
  scale_y_continuous(limits = c(1,3), breaks = c(1,2,3)) +
  coord_flip() +
  theme_bw()
```



TBA Fig. S1E-J

```
# @christian please add code for figures S1E-J
```

Export filtered and normalized RNAseq data

```
saveRDS(RNAseq, file = 'results/bulkRNAseq_mmus_data_filt_norm.rds')
```

```
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
```

```
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] DESeq2_1.28.1               SummarizedExperiment_1.18.2
##  [3] DelayedArray_0.14.1         matrixStats_0.58.0
##  [5] Biobase_2.48.0              GenomicRanges_1.40.0
##  [7] GenomeInfoDb_1.24.2         IRanges_2.22.2
##  [9] S4Vectors_0.26.1            BiocGenerics_0.34.0
## [11] forcats_0.5.1               stringr_1.4.0
## [13] dplyr_1.0.3                 purrr_0.3.4
## [15] readr_1.4.0                 tidyr_1.2.0
## [17] tibble_3.1.4                ggplot2_3.3.3
## [19] tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-6           fs_1.5.0               lubridate_1.7.9.2
##  [4] bit64_4.0.5            RColorBrewer_1.1-2     httr_1.4.2
##  [7] tools_4.0.5            backports_1.2.1        utf8_1.1.4
## [10] R6_2.5.0               DBI_1.1.1              colorspace_2.0-0
## [13] withr_2.4.1            tidyselect_1.1.0       bit_4.0.4
## [16] compiler_4.0.5         cli_2.3.0              rvest_0.3.6
## [19] xml2_1.3.2             labeling_0.4.2         scales_1.1.1
## [22] genefilter_1.70.0      digest_0.6.27          rmarkdown_2.14
## [25] XVector_0.28.0         pkgconfig_2.0.3        htmltools_0.5.2
## [28] highr_0.8              dbplyr_2.0.0           fastmap_1.1.0
## [31] rlang_0.4.10           readxl_1.3.1           rstudioapi_0.13
## [34] RSQLite_2.2.3          farver_2.0.3           generics_0.1.2
## [37] jsonlite_1.7.2         BiocParallel_1.22.0    RCurl_1.98-1.2
## [40] magrittr_2.0.1         GenomeInfoDbData_1.2.3 Matrix_1.3-2
## [43] Rcpp_1.0.7             munsell_0.5.0          fansi_0.4.2
## [46] lifecycle_0.2.0        stringi_1.5.3          yaml_2.2.1
## [49] zlibbioc_1.34.0        grid_4.0.5             blob_1.2.1
## [52] crayon_1.4.0           lattice_0.20-41        splines_4.0.5
## [55] haven_2.3.1            annotate_1.66.0        hms_1.0.0
## [58] locfit_1.5-9.4         knitr_1.31             pillar_1.6.2
## [61] geneplotter_1.66.0     reprex_1.0.0           XML_3.99-0.5
## [64] glue_1.4.2             evaluate_0.14          modelr_0.1.8
## [67] vctrs_0.3.8            cellranger_1.1.0       gtable_0.3.0
## [70] assertthat_0.2.1       cachem_1.0.3           xfun_0.31
## [73] xtable_1.8-4           broom_0.7.4            survival_3.2-7
## [76] AnnotationDbi_1.50.3   memoise_2.0.0          ellipsis_0.3.2
```